Art Unit 232 Paper No. 36

Appeal No. 90-2231 rpb

ON BRIEF

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte Martin G. Reiffin

Application for Patent filed April 3, 1985, Serial No. 06/719,507, which is a continuation-in-part of Serial No. 06/425,612, filed September 28, 1982. Computer System With Real-Time Compilation.

Martin G. Reiffin, pro se.

Primary Examiner - Raulfe B. Zache et al.[1]

Before Hairston, Krass and Martin, Examiners-in-Chief.

Martin, Examiner-in-Chief.

This is a decision on an appeal under 35 U.S.C. § 134 from the final rejection of claims 51, 54 and 56. Claim 53 has been allowed and claims 52 and 55 have been objected to for depending on rejected claims 51 and 54, respectively (answer, ¶¶ 5 and 6).

The subject matter of the invention is readily apparent from representative claim 51:

[1] Hereinafter the examiner.

51. A computer system for the concurrent entry and processing of code, and comprising

a console having keys for entry of successive characters constituting said code,

a memory storing a buffer for holding said code, a code processor program consisting of a sequence of machine instructions, an interrupt service routine at a predetermined memory address, and a stack,

a central processing unit for executing said code processor program instructions and having an interrupt input, a program counter for storing the memory address of the next instruction to be executed by the central processing unit, and means immediately responsive to activation of the interrupt input to push the contents of the program counter onto said stack and to load into said program counter said memory address of the interrupt service routine,

means to cause the central processing unit to execute said code processor program instructions whereby the code processor program thereafter continuously maintains control of the central processing unit, except during interrupts, so as to process the code concurrently as the code is being entered into the system,

means responsive to striking one of said keys to activate said interrupt input whereby the contents of the program counter are pushed onto the stack and the memory address of the interrupt service routine is loaded into the program counter thereby immediately passing control of the central processing unit to the interrupt service routine, whereupon the code processor program is interrupted immediately upon completing the machine instruction which it was executing when the key was struck,

said interrupt service routine including an editor program for inserting into said buffer a character code corresponding to said struck key, and having a return instruction at the end thereof,

said central processing unit having means responsive to execution of said return instruction to pop from the stack into the program counter the memory address of the code processor program instruction immediately following the instruction interrupted by striking said key, whereby the code processor program continues processing said character codes at the location in the buffer where the code processor program was interrupted.

The references relied on by the examiner are:

Catiller et al. (Catiller)	4,293,909	Oct. 6, 1981
Lawrence et al. (Lawrence)	4,464,730	Aug. 7, 1984
Maddock	4,513,391	Apr. 23, 1985 (filed Mar. 19, 1982)

Claim 56 stands rejected under § 102(e) as being anticipated by Lawrence.

Claims 51 and 54 stand rejected under § 103 as unpatentable for obviousness over Lawrence considered alone or in view of Maddock and Catiller.

Reference may be made to the answers and the briefs[2] for the details of the examiner's and the appellant's positions.

OPINION

For the following reasons, we will sustain the § 102(e) rejection of claim 56 but not the § 103 rejection of claims 51 and 54.

We will first address claims 51 and 54. Regarding the meaning of the terms "interrupt" and "interrupt service routine" in these claims, we agree with appellant (opening brief, p. 27) that "when a first routine is interrupted by a second routine, the latter gets control of the central processing unit _immediately_, and not after some delayed subsequent event . . ." (emphasis appellant's). See also the Third Reiffin Aff., ¶ 16 (paper No. 24). That is, the interrupt routine gets control of the central processing unit immediately following execution of the current machine code instruction. This is also consistent

[2] Inasmuch as the reply brief (paper No. 27), filed January 23, 1989, was denied entry by the examiner (paper No. 30), we have not considered it.

with Oppenheim's description of an interrupt service routine (Oppenheim Aff., ¶ 22):[3]

> An interrupt service routine, by definition, gets control immediately upon completion of the machine code instruction being executed at the instant of the interrupt. This is a period measured in microseconds. . . . The only circumstance that can delay passing control to the interrupt service routine is the present activation of a different interrupt having a higher priority.

In addition to calling for an "interrupt service routine," claims 51 and 54 further specify that the interrupt service routine "includ[es] an editor program."

In our opinion, Lawrence, Maddock and Catiller do not disclose or suggest an interrupt service routine which includes an editor program.

Considering Lawrence first, we agree with appellant that the text editing function performed by text editor 10 in the Fig. 1 embodiment (as well as by microprocessor 26 in the Fig. 5 embodiment and apparently by data processor 24 in the Fig. 4 embodiment) is not part of an interrupt service routine. This is readily apparent from the fact that termination of formatting in response to a keystroke, which in turn invokes the text editor (col. 3, lines 8-9), occurs at the end of the current row of displayed text (col. 11, lines 51-59) rather than immediately following execution of the current machine code instruction.

[3] This affidavit accompanied appellant's Supplemental Reply Brief (paper No. 35).

Nor is there anything else in the Lawrence specification which suggests that the text editor can be part of an interrupt service routine. Although Lawrence's claim 4 was amended during prosecution to specify that "said editing means includes means to interrupt said formatting means upon receipt of a keystroke from said keyboard invoking said editing means to perform a text editing operation," we believe that in light of the specification (e.g., col. 11, lines 51-54) the artisan would have construed the term "interrupt" as used in this claim to mean merely that formatting is terminated in response to receipt of a keystroke. In other words, the term "interrupt" in Lawrence's claim 4 would have been given a broader construction than the terms "interrupt" and "interrupt service routine" as used in appellant's claims 51 and 54.

While his text editor is not part of an interrupt service routine, Lawrence does suggest that his apparatus employs an interrupt service routine when, in describing his Fig. 5 embodiment, he identifies line 29 between keyboard adapter 30 and microprocessor 26 as "an interrupt line." Although Lawrence, which is assigned to IBM, does not describe the purpose of this interrupt line, we believe appellant is correct when he contends (opening brief, pp. 29-30; Reiffin Aff., ¶ 19) that the presence of this interrupt line 29 at most suggests the conventional keyboard interrupt routine used in the IBM Personal Computer. This routine, which transfers keyed-in data to a queue buffer where it is held until requested by the CPU, is described in detail in Exhibit A to the Third Reiffin Affidavit and in the exhibits to the Oppenheim Affidavit. We also agree with appellant (opening brief, p. 30; Reiffin Aff., ¶ 19) and Oppenheim (Oppenheim Aff., ¶¶ 10-16) that this keyboard interrupt routine does not include (i.e., perform the functions of) Lawrence's text editor. Nor does it perform any other function which may be considered to be an editing program.

For reasons similar to those given above with respect to Lawrence, Maddock likewise fails to disclose or suggest an interrupt service routine which includes an editor. Column 4, lines 20-28, of Maddock reads as follows:

> During text editing operations, keystrokes from the keyboard 8 are processed by an interrupt handler 14 forming a part of the editor/formatter 7 software to capture the keyed character and put it into a keystroke queue 15 to be processed. The editor thereafter takes keystrokes in order from this queue 15 and directly updates the portion of the document held in the refresh buffer 3, and thus displayed on the screen.

In view of the above, it appears that Maddock's interrupt handler carries out the conventional keyboard interrupt routine used in the IBM Personal Computer and discussed above with respect to Lawrence. Consequently, this keyboard interrupt routine does not perform the functions of Maddock's editor/formatter or any other functions which may be considered to be an editing program.

We have also reviewed Catiller but find therein no suggestion of an interrupt service routine which includes an editor program.

For the foregoing reasons, we will not sustain the examiner's § 103 rejection of claims 51 and 54 as unpatentable for obviousness over Lawrence considered alone or in view of Maddock and Catiller.

On the other hand, we will sustain the § 102(e) rejection of appellant's claim 56, which is broader than claims 51 and 54, as anticipated by Lawrence. The question of whether Lawrence is enabling as to the subject matter relied therein will be addressed below after we identify the subject matter in Lawrence which serves as the basis for this rejection.

Claim 56, in our view, is broad enough to read on any one of Lawrence's Fig. 1, Fig. 4 and Fig. 5 embodiments. Taking as an example the Fig. 5 embodiment, Lawrence employs a console having a keyboard 1 for entry of successive characters constituting code, a random access memory (RAM) 33 which includes a buffer section 36 for storing the entered code and a section 38 for storing an editor (i.e., text editor) program and a code processor (i.e., formatting) program (col. 6, lines 59-64), and a central processing unit (i.e., microprocessor) 26. As explained at column 11, lines 51-54, after the formatting program begins it maintains control of the microprocessor except when a key is struck (thereby invoking the text editor), with the result that the formatting program thereafter (i.e., after operation of the text editor) processes the code concurrently as it is entered into buffer 36. In response to the striking of one of the keys and at the end of the current row, the formatting program passes control to the text editing program, which inserts into buffer 36 a character code corresponding to the struck key.[4] After the character code has been inserted into the RAM, control of the microprocessor is returned to the formatting program, which then resumes formatting the code stored in the buffer. Claim 56 reads on the Fig. 1 and Fig. 5 embodiments in a similar manner.

We note that appellant has not explained how the language of claim 56 distinguishes over Lawrence. Instead, appellant's sole argument for the allowability of this claim is that "the functions recited in this claim are not enabled in the reference" (opening brief, p. 34). More particularly, appellant argues (opening brief, pp. 14-15) that

> the Lawrence et al. patent does not contain an enabling disclosure of either their interpreter/formatter 6 or their text editor 10. These components are "disclosed" in the drawings only as labeled boxes and in the specification only as goals or objects to be achieved. Appellant contends that a box in the drawings labeled "INTERPRETER/FORMATTER" or "TEXT CODE" does not teach one of ordinary skill how to write software or build hardware, nor does a specification which summarily states desired interactions and merely lists the names of functions, goals and objects said to be achieved by unidentified and undisclosed programs.

As noted in In re Wilder, 429 F.2d 447, 166 USPQ 545 (CCPA 1970), a prior publication or patent will not be considered anticipatory

> if it is found not to be sufficiently enabling, in other words, if it does not place the subject matter of the claims within

[4] In contrast to claims 51 and 54, claim 56 does not call for an interrupt service procedure to insert the code into the buffer.

> "the possession of the public." See, e.g., In re LeGrice, 49 CCPA 1124, 301 F.2d 929, 133 USPQ 365 (1962); In re Brown, 51 CCPA 1254, 329 F.2d 1006, 141 USPQ 245 (1964).[5]

This requires that one skilled in the art could have gained possession of the claimed subject matter without undue experimentation. In re Sheppard, 339 F.2d 238, 242, 144 USPQ 42, 45 (CCPA 1964).

Initially, the applicant has the burden of proof of demonstrating that an anticipatory reference patent is non-enabling. In re Sasse, 629 F.2d 675, 681, 207 USPQ 107, 111 (CCPA 1980). If the applicant succeeds in establishing a prima facie case of non-enablement, the burden then shifts to the PTO to rebut the prima facie case. Id.; In re Grasselli, 713 F.2d 731, 738, 218 USPQ 769, 774 (Fed. Cir. 1983).

In our opinion, appellant has not made out a prima facie case of non-enablement as to the text editor 10 or the interpreter/formatter 6 in Lawrence's Fig. 1 embodiment or as to the corresponding structure in the Fig. 4 and Fig. 5 embodiments. In re Donohue, 550 F.2d 1269, 1271, 193 USPQ 136, 137-38 (CCPA 1977), addressed the sufficiency of block diagrams in an application:

> Employment of block diagrams and descriptions of their functions is not fatal under 35 U.S.C. 112, first paragraph, providing the represented structure is conventional and can be determined without

[5] See also In re Donohue, 632 F.2d 123, 125, 207 USPQ 196, 199 (CCPA 1980); In re Borst, 345 F.2d 851, 855, 145 USPQ 554, 557 (CCPA 1965).

> undue experimentation. In re Ghiron, 58 CCPA 1207, 442 F.2d 985, 169 USPQ 723 (1971).

White Consolidated Industries v. Vega Servo-Control, 713 F.2d 788, 791, 218 USPQ 961, 963 (Fed. Cir. 1983), which is relied on extensively by appellant (opening brief, pp. 21-23), similarly states that

> a disclosure is sufficient [under the enablement requirement of § 112, first paragraph] even if it would require that one skilled in the art conduct some experimentation. The amount of experimentation, however, must be reasonable. In re Brandstadter, 484 F.2d 1395, 1404, 179 USPQ 286, 294 (CCPA 1973).

As explained below, appellant has not submitted evidence sufficient to establish that it would have required undue experimentation to implement any of Lawrence's embodiments, let alone all of them, as would be necessary to remove Lawrence as an anticipatory reference.

Appellant contends (opening brief, p. 16) that "the Wordsworth affidavit attempted to explain the Lawrence et al. patent with respect to the issues of anticipation and non-enablement." We assume that the Wordsworth affidavit to which appellant refers is the Third Wordsworth Affidavit (paper No. 19).[6] Although Wordsworth (¶ 29) states that "the purported functions of the Lawrence editor are allegedly implemented by

[6] Wordsworth's first affidavit was filed in parent application Serial No. 06/425,612. His second affidavit, entitled "Supplemental Affidavit," was filed in the instant application (paper No. 7) but makes no reference to Lawrence, which had not yet been applied as a reference.

hardware and/or software neither of which is disclosed in the patent," we see nothing in this affidavit which suggests that Wordsworth believes it would have required undue experimentation for one skilled in the art to implement Lawrence's text editor and interpreter/formatter functions. In fact, the discussion of Lawrence in paragraphs 13-28 indicates that Wordsworth had no difficulty understanding how Lawrence works. For example, he states (¶ 13) that "[t]he Lawrence interpreter/formatter performs all of the display functions of a screen-oriented editor, and together with the rest of the mentioned functions constitutes an editor and only an editor." Furthermore, he indicates (¶ 17) that "[t]he only symbols which the Lawrence interpreter/formatter does analyze and 'interpret' are the editing text commands, such as newline symbols and format commands. But this is exactly what every screen editor has always done . . . ."

Although not relied on by appellant in support of his non-enablement argument, we have also reviewed the Oppenheim affidavit but find therein nothing which suggests that one skilled in the art would have been unable to implement Lawrence without undue experimentation.

In view of the above, the only evidence of record in support of appellant's non-enablement argument is appellant's own affidavit, i.e., the Third Reiffin Affidavit. However, this affidavit, too, is insufficient to establish a <u>prima facie</u> case of non-enablement. Even assuming for the sake of argument that appellant is correct when he states (¶ 8) that the term "interpreter/formatter" "appears to be a coined phrase which is not a term of art in the computer industry" and that "insofar as the appellant is aware, this term is neither defined, explained nor disclosed in any dictionary, treatise, encyclopedia, other patent, or other publication," the test for enablement is not whether this term "identif[ies] any particular program [or] suggest[s] to one skilled in the art any other available program which may be substituted for it," as appellant seems to believe. Rather, the test is whether one skilled in the art, with knowledge of the disclosed functions to be performed by Lawrence's interpreter/formatter and text editor, could have constructed the suitable apparatus without undue experimentation. Thus, as regards Lawrence's Fig. 5 embodiment, for example, the test is whether a programmer of ordinary skill would have been able to write the software for implementing the interpreter/formatter and text editor functions. Appellant has not specifically explained why one skilled in the art would have undue difficulty in writing such software. Nor has appellant explained why the artisan would have had undue difficulty implementing Lawrence's Fig. 1 and Fig. 4 embodiments. As a result, the affidavit is merely a statement of appellants' conclusion that the reference is non-enabling. Affidavits in conclusory form without identification of the particular problems one skilled in the art might be faced with are not sufficient to establish a prima facie case of non-enablement. In re Ludovici, 482 F.2d 958, 965, 179 USPQ 84, 88 (CCPA 1973).

Nor has appellant submitted evidence establishing that it would have taken a computer programmer or circuit designer an unreasonably long time to write software or design hardware sufficient to implement the three embodiments of the Lawrence apparatus. Compare White Consolidated Industries, supra, 713 F.2d at 791, 218 USPQ at 963, wherein a skilled programmer testified that development of a single pass language translator would require from 1-1/2 to 2 man-years of effort, an effort which the court held to be clearly unreasonable.

For the foregoing reasons, we are of the opinion that appellant has failed to make out a prima facie case of non-enablement as to the Lawrence subject matter which is relied on in the § 102(e) rejection of claim 56.

In summary, we have sustained the rejection of claim 56 under § 102(e) as anticipated by Lawrence but have not sustained the § 103 rejection of claims 51 and 54 as unpatentable for obviousness over the teachings of Lawrence, Maddock and Catiller. Accordingly, the decision of the examiner is affirmed-in-part.

No time period for taking subsequent action in connection with this appeal may be extended under 37 CFR 1.136(a). See the final rule notice published at 54 Fed. Reg. 29,548 (July 13, 1989) and 1105 O.G. 5 (Aug. 1, 1989).

AFFIRMED-IN-PART

Kenneth W. Hairston
Kenneth W. Hairston)
Examiner-in-Chief)
)
)
) BOARD OF
Errol A. Krass) PATENT APPEALS
Examiner-in-Chief) AND
) INTERFERENCES
John C Martin
John C. Martin)
Examiner-in-Chief)

Martin G. Reiffin
5439 Blackhawk Dr.
Danville, CA 94526